Astral: a Mixed-Criticality RISC-V SoC Architecture for Satellite Onboard AI

Yvan Tortorella*§, Maicol Ciani*, Riccardo Tedeschi*, Luigi Ghionda*, Andrea Belano*‖, Andrea Fenzi†, Dario Pascucci†, Gianluca Aranci†, Angelo Garofalo*‡, Luca Benini*‡, Davide Rossi*, Francesco Conti*
*University of Bologna, Italy †Thales Alenia Space Italia, Italy
‡ETH Zürich, Switzerland §Fondazione Chips-IT, Italy ‖University of Pavia, Italy

***Abstract*—The integration of AI onboard satellites is a rapidly emerging trend to enable edge data processing, reduce dependence on ground stations, and facilitate faster orbital maneuvers, such as satellite reorientation. This evolution demands a new class of onboard computers with enhanced processing power, real-time control capabilities, and robustness against the harsh conditions of the space environment. We introduce *Astral*, a fully open-source heterogeneous space architecture based on the RISC-V ISA, providing an effective, mixed-criticality template platform for next-generation space-ready SoCs. *Astral* features a dual-core, cache-coherent, Linux-capable host processor based on the open-source CVA6 RV64 core, which can be programmed at runtime to operate in Dual-Core Lockstep (DCLS) mode, enabling fault detection through Double Modular Redundancy (DMR). Additionally, the host processor integrates a dedicated unit for system-level security, and one for AI acceleration using a reconfigurable DMR heterogeneous RISC-V cluster with tensor processing capabilities. The system is further protected by error-correcting codes (ECC) in both the on-chip memory banks and the system interconnect. *Astral* provides $<$ 100 clock cycles recovery from detected faults within the locked cores, 663 GOPS on 3×3 convolution with 8-bit integer precision, 48 GOPS on BFloat16 General Matrix-Matrix Multiplications (GEMM), and $10\times$ boost of Softmax execution over software counterparts. We present preliminary evaluation results based on synthesis in Global Foundries GF12LP+ 12 nm FinFet technology and from physical mapping on AMD Xilinx Ultrascale+ VCU118 evaluation board.**

I. Introduction

The integration of Artificial Intelligence (AI) into satellite systems is a major trend in space technology, transforming satellites into advanced Space Cyber-Physical System (S-CPS) capable of autonomous onboard data processing. This advancement reduces reliance on ground stations and enables rapid decision-making for safety-critical operations such as orbital maneuvers, as well as real-time data analysis. For instance, ESA's Φ-Sat-2 satellite employs onboard AI for real-time image processing [1], [2], ensuring that only essential information is transmitted back to Earth, accelerating decision-making processes and boosting data transmission efficiency.

Implementing AI capabilities onboard requires a new class of satellite computers that provide enhanced processing power while ensuring real-time operation for control function, as well as fault tolerance and resilience against the harsh conditions of space. Traditional satellite operations often rely on human intervention for tasks like orbit control and system maintenance. The integration of AI enhances efficiency and reduces operational costs by enabling autonomous decision-making and real-time data processing. Existing space processors, such as the Cobham Gaisler GR740 [3], are typically based on homogeneous general-purpose computing architectures with limited support for AI acceleration. Moreover, despite being based on open-source ISAs such as RISC-V or SPARC, all available space processors are distributed under restrictive licenses, offering little customization and limiting their use for open research purposes.

This paper introduces *Astral*, a fully open-source[1], heterogeneous, mixed-criticality RISC-V-based space platform designed for developing space-ready System on Chips (SoCs). Leveraging RISC-V's open and modular nature, *Astral* enables extensive system customization and scalability, making it an ideal platform for space applications and space processor research and design exploration.

Astral features a dual-to-quad-core, cache-coherent, Linux-capable host processor, complemented by a RISC-V-based secure subsystem and an AI acceleration domain built upon 8 compact RISC-V cores with dedicated FPUs and application-specific hardware accelerators. To enhance fault tolerance in critical tasks, the system integrates reconfigurable Double Modular Redundancy (DMR) [4] scheme across all processing domains, along with Error-Correcting Codes (ECC)-protected memory, ensuring high-performance processing under normal conditions and system reliability when required. *Astral* provides $<$ 100 clock cycles recovery from faults detected within the locked cores, and performance increase on computationally intensive workloads with 663 GOPS on 3×3 8-bit integer convolution, 48 GOPS on BFloat16 General Matrix-Matrix Multiplication (GEMM), and $10\times$ Softmax boost over software-based execution for modern transformer-based ML models.

The *Astral* architecture provides an easily scalable open-source platform based on area and performance constraints and comes with a ready-to-use development flow for AMD Xilinx Ultrascale+ VCU118 FPGA boards.

II. Architecture

Astral (**A**rchitecture for reliable execution of **S**afety-critical **T**asks based on **R**ISC-V for satellite **A**pp**L**ications) is an open-source architecture for on-board satellite computing. *Astral* is organized in three distinct domains: **host**, **security**, and

[1] https://github.com/pulp-platform/astral

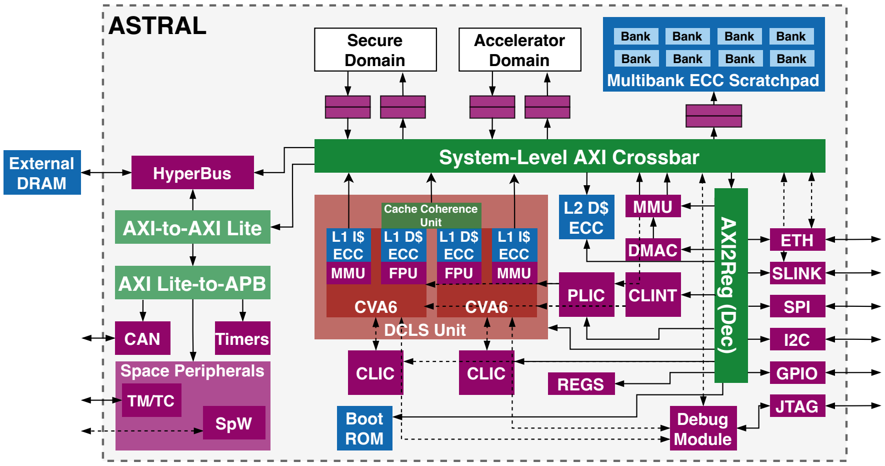

Fig. 1. *Astral* architecture, focused on the dual-core cache-coherent CVA6 **host domain** with DCLS support, ECC scratchpad, and the available peripherals.

accelerator – each designed to provide essential features for emerging space applications.

A. *Host domain*

Figure 1 illustrates the *Astral* host domain architecture, designed to support full-fledged operating systems and complex software running directly onboard. The host domain relies on the OpenHW Group CVA6. CVA6 is a RISC-V application-class, single-issue, in-order processor with six pipeline stages implementing the RV64G Instruction Set Architecture (ISA) with configurable hypervisor extension, fast interrupt virtualization via dedicated Core Level Interrupt Controller (CLIC), and TLB partitioning to boost its real-time capabilities. *Astral*'s host domain includes a dual-core CVA6 cluster with snooping-based coherent L1 data caches compliant with the ACE protocol [5], as well as a unified L2 data cache and an on-chip SRAM scratchpad. To enhance the mixed-criticality and fault tolerance capabilities of the *Astral* host domain, the two CVA6 cores are coupled with a runtime-programmable hybrid modular redundancy unit [4] allowing the two cores to switch between two operating modes: a *performance* mode, where the two CVA6 cores act as independent parallel processors operating on different data and instructions; and a *redundant* mode. In the latter, the two CVA6 cores are grouped to execute code in DCLS operation, exploiting a DMR scheme. To switch to redundant mode, the two CVA6 cores rely on a fence-based synchronization mechanism, after which they are fed with the same inputs (in terms of instructions, fetched data, and interrupt lines) and cycle-by-cycle produce the same outputs (e.g., data stored in memory) and internal state. To guarantee result consistency, the hybrid redundancy unit provides an online checker that continuously compares the output and state of the two cores and, in case of mismatch, triggers a recovery mechanism that restores the cores state within 100 clock cycles. Furthermore, the on-chip L2$ and Scratchpad are protected with Single Error Correction, Double Error Detection (SEC-DED) ECC encoding, with interrupt signaling in case of multiple error occurrence.

The *Astral* host domain is organized around a system-level AXI crossbar and integrates a RISC-V-compliant Core-Local Interrupt Controller (CLINT), a Platform-Level Interrupt Controller (PLIC), an I/O Memory Management Unit (IOMMU), a high-performance system-level Direct Memory Access Controller (DMAC), and several memory-mapped IPs, including Ethernet, SPI, and I2C. Additionally, it supports external DRAM via a single-PHY HyperBus controller. The host domain also includes a boot ROM and a debug module accessible via the JTAG interface. To demonstrate its flexibility as an open-source platform, the proposed *Astral* configuration additionally incorporates Telemetry and Telecommand (TM/TC) [6], [7] and SpaceWire [8] peripherals from Thales Alenia Space Italy.

B. *Secure domain*

The secure domain, shown in Figure 2a, provides secure boot and hardware Root-of-Trust (RoT) services for *Astral*. It is based on the open-source OpenTitan project in the *Earl Grey* variant, modified to integrate seamlessly into the *Astral* SoC via the system AXI interconnect, accessible only through a mailbox-based message exchange to safeguard system security [9]. The *Astral* secure domain includes a dual-core lockstep Ibex RISC-V processor (RV32IMC, in-order, two-stage pipeline) with delayed execution between the cores as an anti-tampering countermeasure. It also features a set of hardware accelerators for cryptographic operations, namely AES, HMAC (SHA-256), and KMAC (SHA-3). Additionally, it features OTBN, a coprocessor dedicated to asymmetric encryption, essential for key exchange and digital signatures. The secure domain can also be configured to include secure SRAM, embedded Flash, ECC, and eFuse-based key storage, along with a key manager for hardware identity protection.

C. *Accelerator domain*

The third main component of *Astral* is the accelerator domain, shown in Figure 2b, designed to enhance onboard

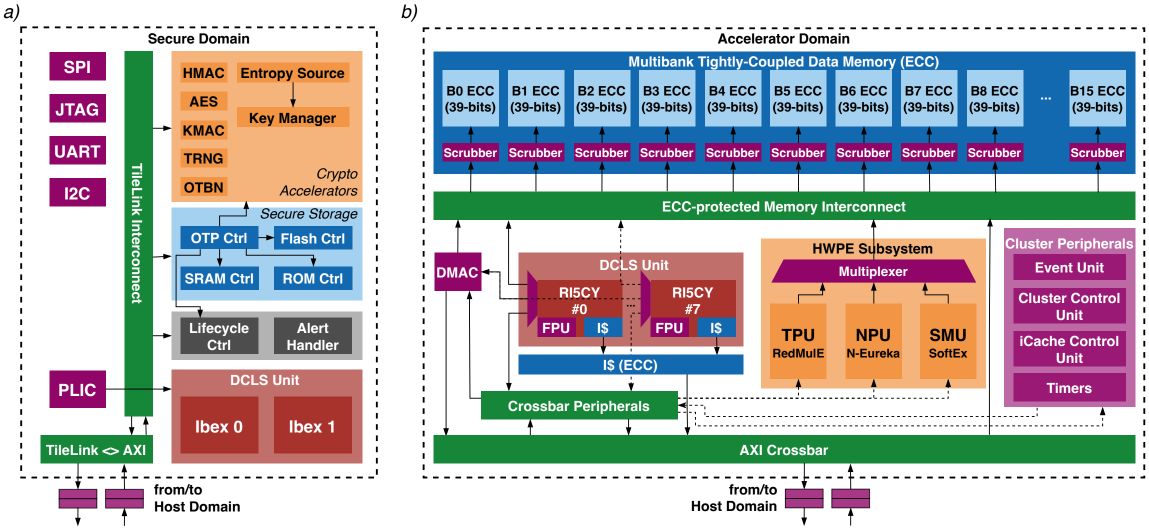

Fig. 2. *Astral's a)* **secure domain** with two Ibex cores in DCLS configuration; *b)* **accelerator domain** with 8 RI5CY DSP cores configurable in DCLS mode, ECC TCDM, and three HWPEs: TPU, NPU, and SMU.

machine learning and AI performance. Based on the open-source Parallel Ultra-Low-Power (PULP) cluster in a highly heterogeneous configuration, it features 2–16 RISC-V digital signal processing cores based on the RI5CY architecture (RV32IMCFXpulpV2, in-order, four-stage pipeline) with dedicated FPUs, a hierarchical instruction cache, and a DMAC for efficient data transfer across the memory hierarchy. RI5CY cores are wrapped by a hybrid redundancy unit that can be reconfigured at runtime to make the cores operate in DCLS mode for reliable processing in space. In this mode, the processors are grouped into four pairs, each exploiting DMR. The accelerator domain includes a single L1 TCDM, organized into 16–32 banks, each with a 32-bit data width, shared among all the available computing entities and protected by SEC-DED ECC encoding, supported by programmable memory scrubbers. A low-latency, high-bandwidth heterogeneous cluster interconnect (HCI) enables the DSP cores to share the TCDM with three domain-specific Hardware Processing Enginess (HWPEs), which further enhance performance for specific applications. The first of these accelerators is a highly parametric Tensor Processing Unit (TPU), based on RedMulE, accelerating 16-bit (FP16/BFloat16) and 8-bit (E4M3/E5M2) floating-point GEMM and other matrix operations (GEMM-Ops) [10]. Then, a SoftMax Unit (SMU), based on SoftEx, provides fast computation for softmax and GELU kernels in 16-bit floating-point precision (FP16/BFloat16) [11]. Finally, a scalable Neural Processing Unit (NPU), based on N-EUREKA, accelerates deep neural network layers with 2–8-bit weights and 8-bit activations [12]. Such accelerators are highly scalable in terms of number of istances and number of internal computing blocks, spanning from very tiny accelerators for energy-efficient edge processing, to large computing units for increased performance applications. The (HCI) interconnect is then extended with Hsiao SEC-DED ECC redundancy scheme across all communication paths between the memory banks and the available computing units (i.e. cores and HWPEs) to increase the resilience to faults during memory accesses.

III. Experimental Results

A. ASIC Implementation

We implemented the *Astral* architecture in synthesizable SystemVerilog, with IPs from the PULP and OpenTitan open-source projects. We targeted GlobalFoundries GF12LP+ 12nm technology to evaluate the proposed architecture in terms of area and performance, using Synopsys Design Compiler for synthesis and Cadence Innovus for place&route. For these evaluations, we consider a host domain with 16KiB of L1 I$ and 32KiB of L1 D$ for each CVA6 core, plus 128KiB for the L2 D$. The host scratchpad is 128KiB organized in 16 banks. For the accelerator domain, we opted for an NPU configuration with 16 Processing Elements (PEs), each with 288 1×8-bit multipliers and 32 accumulators; a TPU with 12×4 8-to16-bit Fused-Multiply-Add units; and a SMU with 16 lanes. The secure domain uses the same modified *Earl Grey* configuration presented in *Ciani et al.* [9].

As shown in Figure 3, the host domain occupies 49.5% of the *Astral* architecture, while the accelerator and secure domains account for 27.3% and 23.4%, respectively. Within the host domain, the CVA6 cores (including caches and a redundancy unit) occupy 29% of the area, while the L2 cache and scratchpad memory take up 32%. The secure domain consists mainly of on-chip memory (58%), cryptographic accelerators (22%), and Ibex cores (6%). Other components collectively account for 9%.

In the accelerator domain, the on-chip scratchpad takes 31%, while the HWPE subsystem occupies 41%, with contributions from the NPU (27%), TPU (10%), and SMU (4%). The RI5CY cores account for 11%, while the programmable redundant unit for 2%. Performance-wise, the NPU achieves up to 662 GOPS for 3×3 convolutions and 196 GOPS for

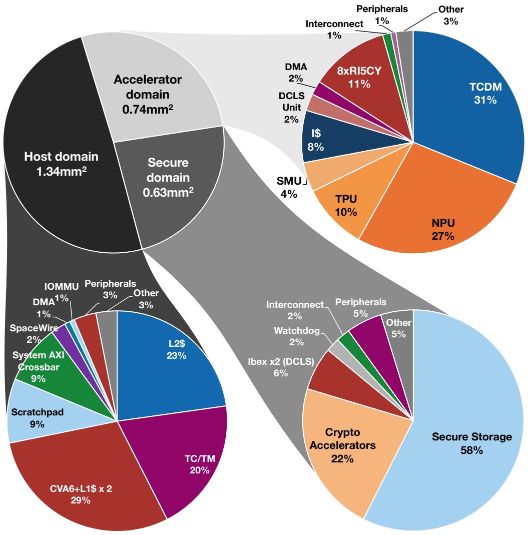

Fig. 3. Detailed distribution of post-synthesis area of *Astral* architecture in the reported configuration.

1×1 convolutions/matrix multiplications in 8-bit precision. The TPU delivers 48 GOPS in 8/16-bit FP precision with 99.4% MAC utilization on a 96×96 tensor. The SMU is 10× faster than optimized software, computing Softmax with sequence length 512 in $\sim 130k$ cycles (~ 2 cycles per element).

Post-layout, *Astral* achieves timing closure for all domains with $\sim 74\%$ density for the secure domain, $\sim 62\%$ for the accelerator, and $\sim 55\%$ for the host.

B. FPGA Implementation

Besides the ASIC implementation, *Astral* provides a plug-and-play FPGA development flow, targeting the AMD Xilinx UltraScale+ VCU118 board. As this board is too small to accommodate the entire *Astral* design, we synthesized two separate architectural configurations: one incorporating the host domain and the accelerator domain and the other featuring the host domain and secure domain. Both can rely on the DDR4 controller available on the board or on a more compact HyperBus accessing the HyperRAM chips through an FMC connector[2]. We targeted 50 MHz and 20 MHz frequency for the host and accelerator/secure domains repsectively and show the implementation results, in terms of resource utilization – Look-up-Tables (LUTs), Flip-Flops (FFs), and memory tiles (BRAM, URAM) – in Table I.

IV. Conclusion

We presented preliminary implementation results for the *Astral* platform, a fully open-source architecture for on-board satellite computing.

Acknowledgment

We thank Thales Alenia Space Italia and Fondazione Chips-IT, Italy, for sponsoring the project, and the Design Zentrum of ETH Zürich for the technical support.

[2] https://github.com/pulp-platform/fmc_peripheral_boards

TABLE I
AMD Xilinx UltraScale+ VCU118 Resource Utilization Table

	LUTs	FFs	BRAM	URAM
Astral	**1.23M**	**655114**	**366.5**	**20**
Host Domain	**389841**	**299395**	**276**	**4**
CVA6+L1$+CLIC (x 2)	140102	63135	204 (L1$)	-
Host Memory	59204	33252	64 (L2$)	4 (Scratch.)
IOMMU	6483	5583	-	-
TC/TM	49277	100970	8	-
Spacewire	14088	19680	-	-
Others (Peripherals etc.)	120687	76775	-	-
Secure Domain	**249739**	**146737**	**40**	**16**
Ibex x2	27362	9685	7	-
Crypto Accelerators	83222	47828	13.5	-
Secure Storage	35235	16882	19.5	16
Others (Peripherals ecc.)	103920	72342	-	-
Accelerator Domain	**590537**	**208982**	**50.5**	**-**
RI5CY (x 8)	115400	35384	-	-
DCLS Unit	33554	8305	-	-
TCDM	8243	1152	40	-
HWPEs	351360	98336	-	-
Others	81980	65805	10.5 (I$)	-

References

[1] ESA, "Φsat-2 – Enhancing onboard AI processing," 2024.

[2] ESA, "New satellite demonstrates the power of AI for Earth observation," 2024.

[3] M. Hijorth, M. Aberg, N. J. Wessman, J. Andersson, R. Chevallier, R. Forsyth, R. Weigand, and L. Fossati, "GR740: Rad-Hard Quad-Core LEON4FT System-on-Chip," in *Programme and Abstracts Book of the DASIA 2015 Conference*, vol. 732. Barcelona, Spain: ESA, Sep. 2015, p. 7, conference Name: DASIA 2015 - Data Systems in Aerospace ADS.

[4] M. Rogenmoser, Y. Tortorella, D. Rossi, F. Conti, and L. Benini, "Hybrid Modular Redundancy: Exploring Modular Redundancy Approaches in RISC-V Multi-core Computing Clusters for Reliable Processing in Space," *ACM Trans. Cyber-Phys. Syst.*, vol. 9, no. 1, pp. 8:1–8:29, Jan. 2025.

[5] R. Tedeschi, L. Valente, G. Ottavi, E. Zelioli, N. Wistoff, M. Giacometti, A. B. Sajjad, L. Benini, and D. Rossi, "Culsans: An efficient snoop-based coherency unit for the cva6 open source risc-v application processor," 2024.

[6] "CCSDS 131.0-B-5 - TM Synchronization and Channel Coding | The Consultative Commitee for Space Data Systems," 2023.

[7] "CCSDS 231.0-B-4 | TC Synchronization and Channel Coding. Issue 3. Recommendation for Space Data System Standards (Blue Book)," 2021.

[8] "ECSS-E-ST-50-12C Rev.1 – SpaceWire – Links, nodes, routers and networks (15 May 2019) | European Cooperation for Space Standardization," 2019.

[9] M. Ciani, E. Parisi, A. Musa, F. Barchi, A. Bartolini, A. Kulmala, R. Psiakis, A. Garofalo, A. Acquaviva, and D. Rossi, "Unleashing OpenTitan's Potential: A Silicon-Ready Embedded Secure Element for Root of Trust and Cryptographic Offloading," *ACM Trans. Embed. Comput. Syst.*, Sep. 2024.

[10] Y. Tortorella, L. Bertaccini, L. Benini, D. Rossi, and F. Conti, "RedMule: A mixed-precision matrix–matrix operation engine for flexible and energy-efficient on-chip linear algebra and TinyML training acceleration," *Future Generation Computer Systems*, vol. 149, pp. 122–135, Dec. 2023.

[11] A. Belano, Y. Tortorella, A. Garofalo, L. Benini, D. Rossi, and F. Conti, "A Flexible Template for Edge Generative AI with High-Accuracy Accelerated Softmax & GELU," Dec. 2024.

[12] A. Prasad, L. Benini, and F. Conti, "Specialization meets Flexibility: A Heterogeneous Architecture for High-Efficiency, High-flexibility AR/VR Processing," in *Proceedings of the 2023 Design Automation Conference (DAC 2023), to Appear*, 2023.